This summary is meant to convey preliminary ideas for the purpose of getting feedback. It does not necessarily represent the consensus of the members of the session.
<table>
<thead>
<tr>
<th>Name</th>
<th>Organization</th>
<th>Name</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike Gagliardi</td>
<td>SEI</td>
<td>Rolf Siegers</td>
<td>Raytheon</td>
</tr>
<tr>
<td>Bill Wood</td>
<td>SEI</td>
<td>Don O'Connell</td>
<td>Boeing</td>
</tr>
<tr>
<td>Amine Chigani</td>
<td>Virginia Tech</td>
<td>Jeffery Kertis</td>
<td>Bettis.gov</td>
</tr>
<tr>
<td>David Mason</td>
<td>US ARMY – CECOM SEC</td>
<td>Dave Zubrow</td>
<td>SEI</td>
</tr>
<tr>
<td>Chi-Fang Chen</td>
<td>Boeing</td>
<td>Bob Krut</td>
<td>SEI</td>
</tr>
<tr>
<td>Rick Kazman</td>
<td>SEI</td>
<td>Brian Boucier</td>
<td>Northrop-Grumman</td>
</tr>
<tr>
<td>John Bergey</td>
<td>SEI</td>
<td>Ibrahim Habli</td>
<td>CS.York.ac.uk</td>
</tr>
<tr>
<td>Richard Anderson</td>
<td>US ARMY</td>
<td>Rob Wojcik</td>
<td>SEI</td>
</tr>
<tr>
<td>Jack Li</td>
<td>US ARMY – CECOM SED</td>
<td>Tim Morrow</td>
<td>SEI</td>
</tr>
<tr>
<td>Ron Koontz</td>
<td>Boeing</td>
<td>Nanette Brown</td>
<td>Pitney-Bowes</td>
</tr>
<tr>
<td>Bob Ellison</td>
<td>SEI</td>
<td>Daniel Pipitone</td>
<td>SEI</td>
</tr>
</tbody>
</table>
Why is this important?

Software intensive systems often suffer severe integration and operational/behavioral problems due to lack of consistency between the system and software architectures in addressing system quality attributes.

Oftentimes resulting in costly re-architecting/re-design efforts and operational failures.

This significantly impacts system cost, schedule and mission effectiveness.
Current State of Practice - 1

Technical – 1

- Quality Attribute (QA) requirements and specifications
  - Underspecified quality attribute requirements
  - Difficult to identify requirements and quality attributes omissions at the system level.

- System architecture representations, analysis techniques, tactics, etc.
  - Component relationships at the system architecture level don’t have the robustness that occurs at the software level.
  - Lack of description of tactics at the system level for many QAs.
  - Lack of analytical support in many instances for QAs
  - No real techniques are available to look at quality attributes beyond the “well known.”
  - Systems architecture = system hardware block diagrams. Little notion of multiple system views (h/w maintainability, sustainability, etc)
  - DoDAF doesn't incorporate software concerns. Level of abstractions hurt the communication.
Current State of Practice - 2

Technical - 2

• **Metrics**
  - Lack of metrics about other important quality attributes. Increases difficulty in reasoning about these QA’s.

• **Common semantics**
  - No common language, information models between system and software groups. (Have software people attend system design meetings – cross pollination.)
  - System architects understand quality attributes but in different terminology

• **Architecturally significant scenarios**
  - Software is growing towards scenario based but this creates a bigger disconnect with some system folks.
  - Mission threads are a good starting point for system scenarios (sustainment, availability, performance)
  - Used scenario templates in system fault tree analysis activities.
Current State of Practice - 3

Process
• System architecture level decisions often pushed down to lower software architecture levels
• Lack of communication between system and software groups.
• System architecture the result of a water fall process with functional decomposition
• System architecture folks don’t fully understand/address all of the quality attributes.
• Early and often integration and test is what saves them, but this may not fully scale up.
• System architecture typically defined before making hardware/software trade-offs.

Programmatic
• Time and cost constraints make it difficult to get to the Quality Attributes.
• Customers drive decision making process. Need architecture centric acquisition strategies.
• Lack of ROI data impacts level of commitment and ability to make the sales pitch to management.
• Pressure to allow prototypes to become products.
• System engineering (and acquisition) community lack of awareness of QAW and ATAM's.
• System Engineers/Architects need to be trained and added to ATAM evaluation teams

Organizational/Cultural
• Separation of systems and software groups often occurs.
• Large software groups (5X) compared to system groups.
• No real system architecting team.
• Lack of respect between the system and software group. Parallel learning experiences create different cultures.
• Post mortem activities are the current means for improvement, but rarely get applied.
Technical Gaps and Issues

• Gaps
  – Quality attribute requirements and specifications
  – System architecture representations, analysis techniques, tactics, etc
  – Quality attribute metrics
  – Common semantics
  – System scenarios

• Issues
  – Should architecture work be centralized to all of the work that goes on for System of Systems (SoS) development? Strict, centralized system of system architect/architecture concept in question in this context.
  – Can ATAM be moved into the system area? Concerned about scale, decomposition, scenario/mission thread, schedule and other issues
  – SOA and web services impacts upon architecture and vice-versa are not well-understood.
  – Need to think about intersystem policies as the next higher level of abstraction. However, technology may be moving too fast to cover policies
  – Open source impact.
Recommendations
(developed by SEI)

• **Near-Term**
  – Collaborate with external organizations to understand their needs and any extensions to QAW, ATAM, etc for System Architecture.
    • Identify necessary extensions to existing SAT methodologies.
    • Initiate Pilots and Case Studies with External Collaborators
  – Interview individual participants
  – Put system engineers/architects through the SAT curriculum and follow-up.

• **Long-Term**
  – Active participation in the following to include QA concerns:
    • INCOSE to gain more insight and influence their architecture framework activities.
    • Sys ML standardization and Quality Attribute UML extensions.
    • DoDAF, OMG, TOGAF, etc.
  – Describe tactics, patterns, etc for the remaining important QAs and at the system level. Collaboration with external organizations is necessary.
  – System Scenarios using Mission Threads are a good starting point:
    • Augment for all important QAs
    • Transform into software specific scenarios
    • Collaboration with external organizations is necessary. Pilot activities needed.
  – Investigate how SOA contributes/impacts operational to s/w transformations
  – Collaborate with external organizations and SEMA regarding QA metrics.
  – Leverage the academic community to begin the process of getting methodologies such as QAW and ATAM into the systems engineering process.
  – Figure out what to do about addressing common semantics (technical and cultural issue)
  – Investigate System of Systems architecture role, activities, etc… based on a more decentralized approach at the SoS level.